

MODUL I

Pengenalan Environment OOP

I. TUJUAN

- Memahami lingkungan dasar NetBeans IDE serta cara meng-compile dan menjalankan program
- Membandingkan pemrograman terstruktur dengan pemrograman berorientasi objek dengan membuat program dalam bahasa C++ dan Java.

II. TOOL

- CodeBlocks / Borland C++ / Visual C++
- NetBeans IDE 8.1
- Java SE Development Kit 8

III. DASAR TEORI

a. Pengantar Pemrograman Berorientasi Objek

Apa itu OOP ? Merupakan teknik membuat suatu program berdasarkan objek dan apa yang bisa dilakukan objek tersebut. Object-oriented program terdiri dari objek-objek yang berinteraksi satu sama lain untuk menyelesaikan sebuah tugas.

Kenapa menggunakan OOP ? Kode-kode di-breakdown agar lebih mudah di-manage. Breakdown berdasarkan objek-objek yang ada pada program tersebut. Dianjurkan diimplementasikan untuk program dengan berbagai ukuran karena lebih mudah untuk men-debug.

Pemrograman procedural mengatur program dalam barisan-barisan linier yang bekerja dari atas ke bawah. Kumpulan tahapan yang dijalankan setelah yang lain berjalan. Baik untuk program kecil yang berisi sedikit code. Tidak dianjurkan diimplementasikan pada program berukuran besar, karena susah untuk dimanage dan di-debug

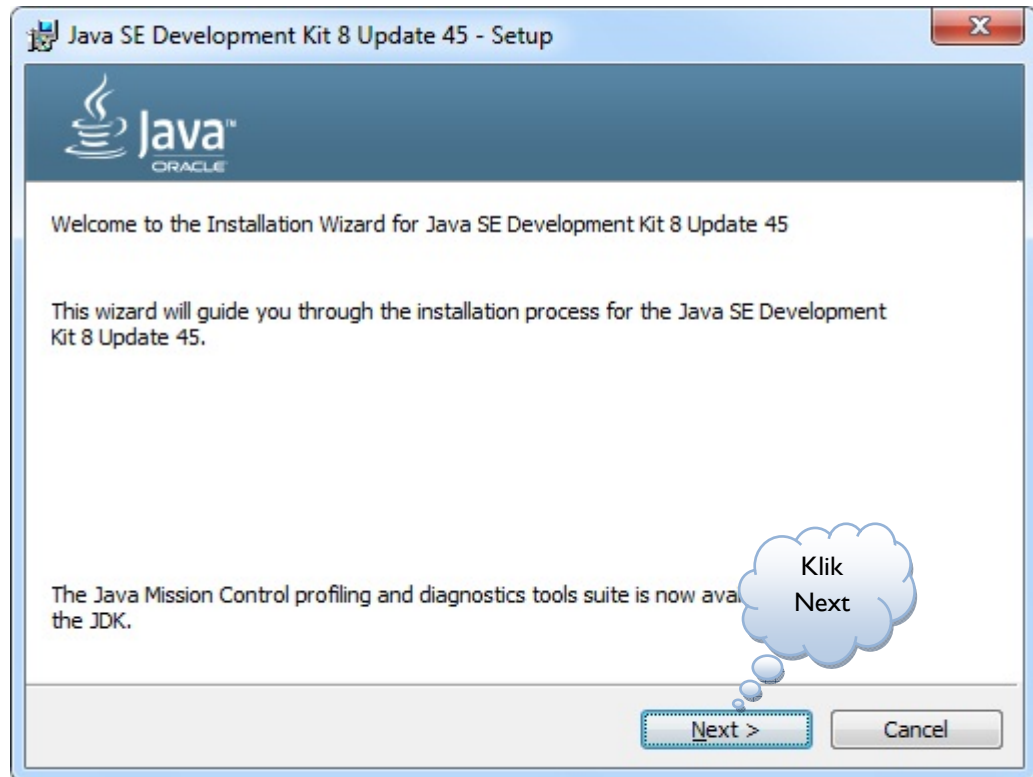
OOP mempunyai karakteristik sebagai berikut :

- Abstraksi
Menemukan hal-hal yang penting pada suatu objek dan mengabaikan hal-hal yang sifatnya insidental.
- Enkapsulasi
Peng kapsulan adalah proses pemaketan data objek bersama method-methodnya
- Pewarisan (Inheritance)
Proses penciptaan kelas baru (subclass/kelas turunan) dengan mewarisi karakteristik dari kelas yang sudah ada (superclass/kelas induk), ditambah karakteristik unik kelas baru itu.
- Reuseability
Reuseability adalah kemampuan untuk menggunakan kembali kelas yang sudah ada.
- Polymorphism
Polymorphism berasal dari bahasa Yunani yang berarti banyak bentuk.

b. Instalasi Java SE Development Kit 8

Langkah-langkah Instalasi Java SE Development Kit 8 :

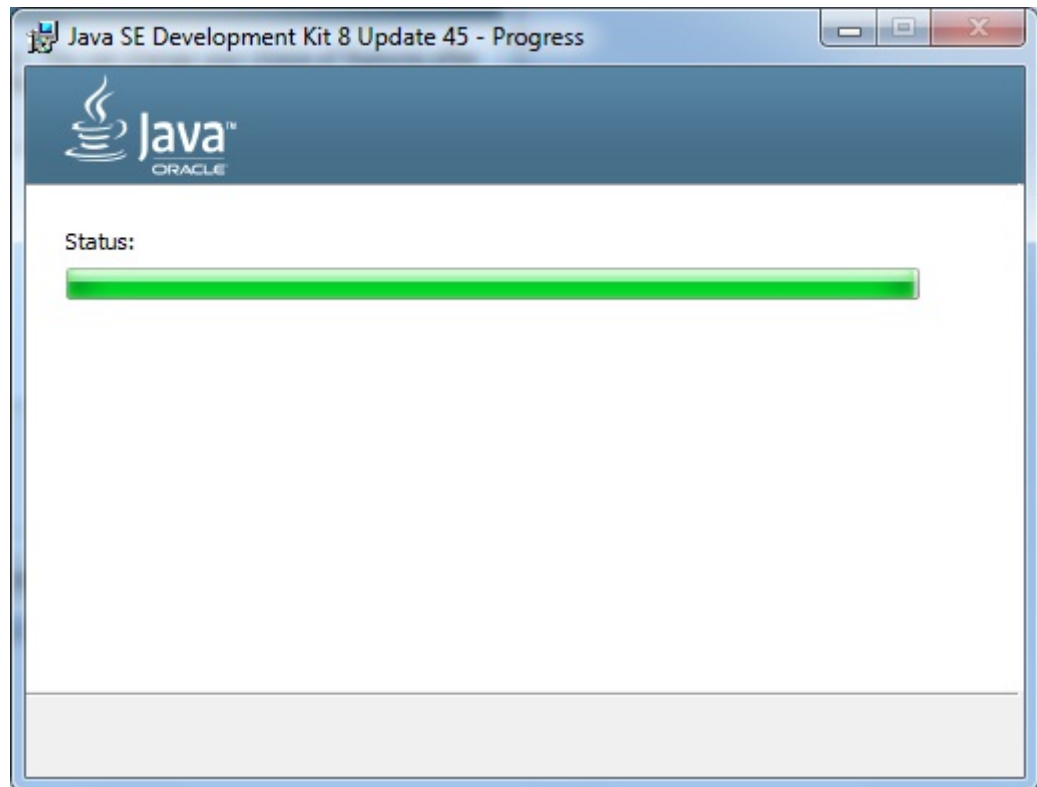
1. Pilih File Setup, kemudian akan muncul tampilan sebagai berikut :



Gambar 1. Java SE Development Kit 8 Setup



Gambar 2. Java SE Development Kit 8 Custom Setup

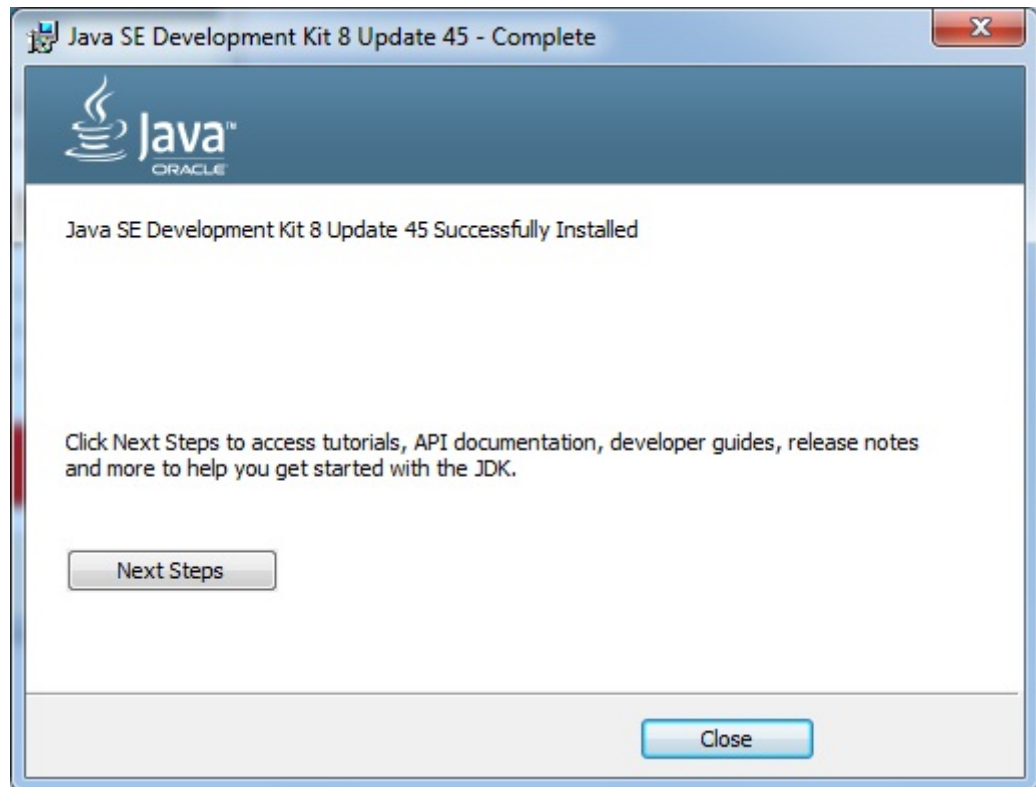


Gambar 3. Java SE Development Kit 8 Custom Setup

2. Pilih Destination Folder dimana JDK tersebut mau diinstallasi, seperti pada tampilan sebagai berikut :



Gambar 4. Java SE Development Kit 8 Custom Setup

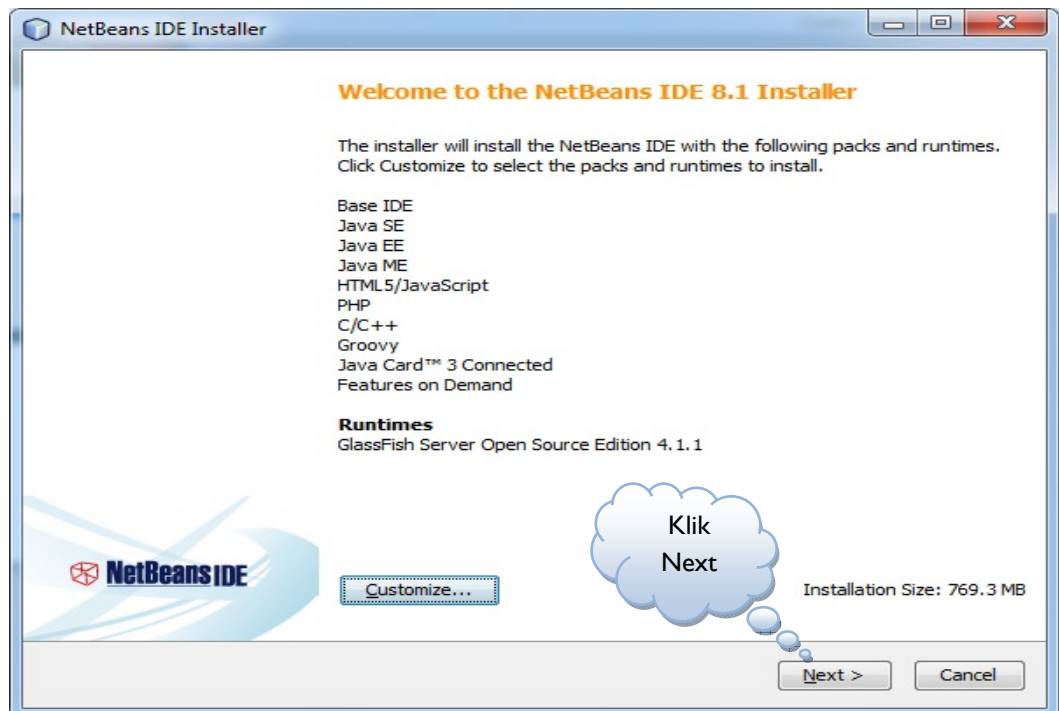


Gambar 5. Java SE Development Kit 8 Complete

c. **Instalasi NetBeans IDE 8.1**

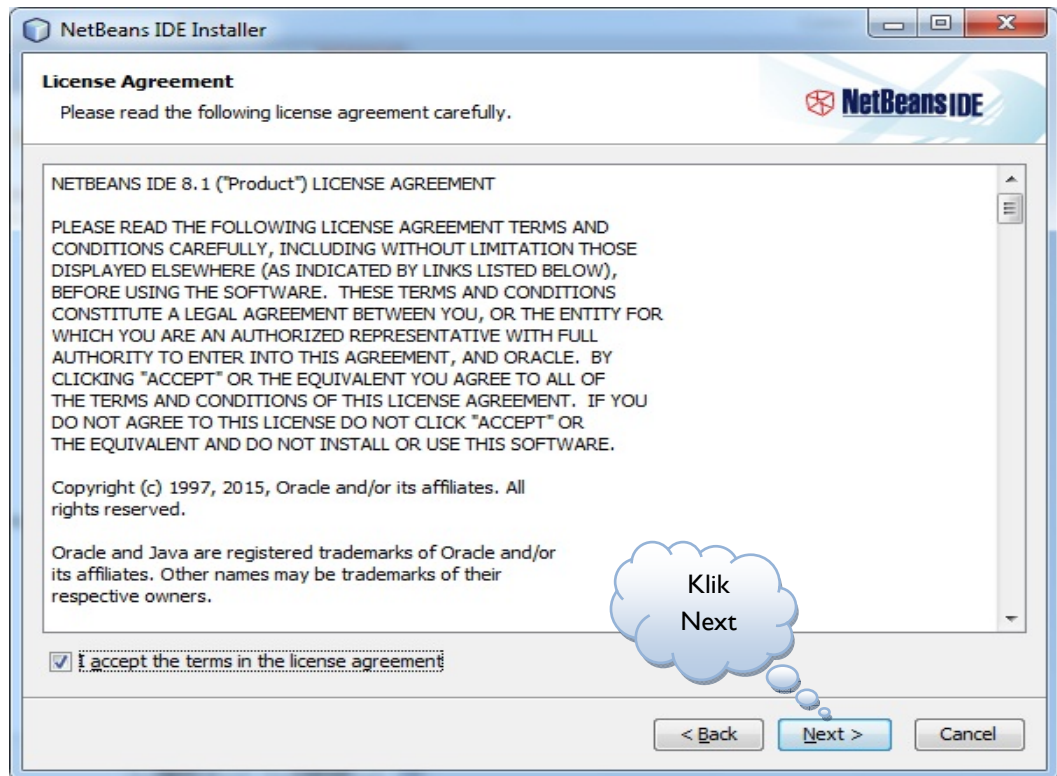
Langkah-langkah Instalasi NetBeans IDE 8.1 :

1. Pilih File Setup, kemudian akan muncul tampilan sebagai berikut :

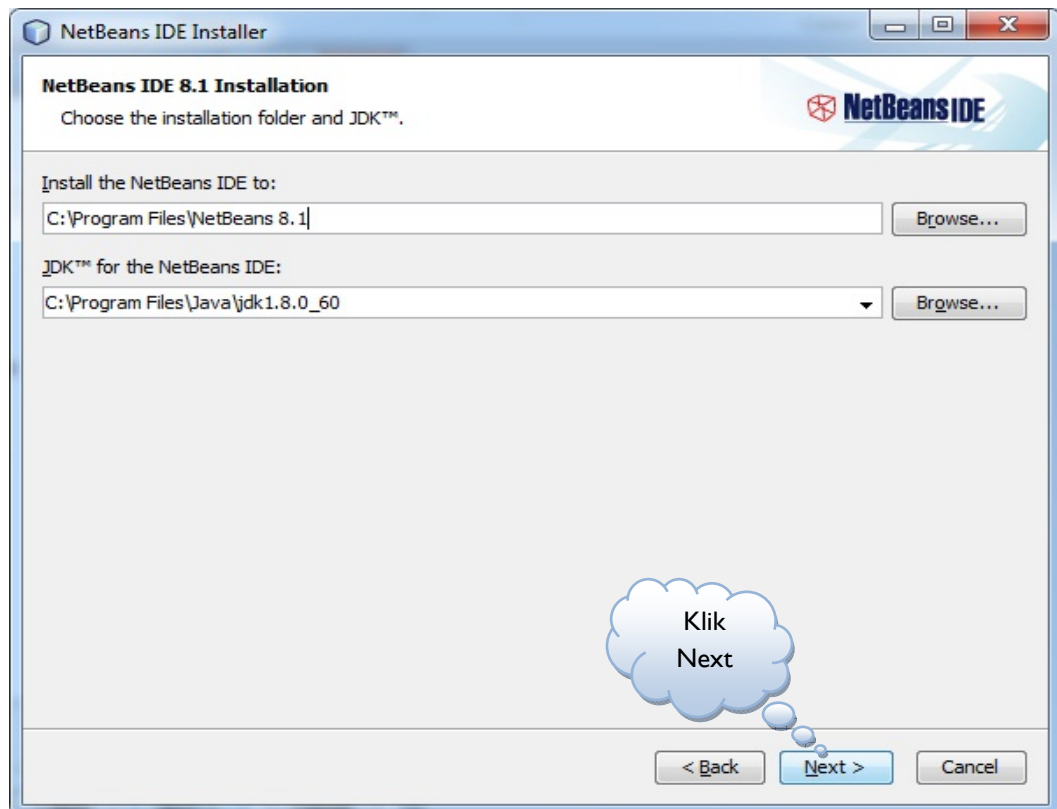


Gambar 6. Wellcome NetBeans IDE Installer

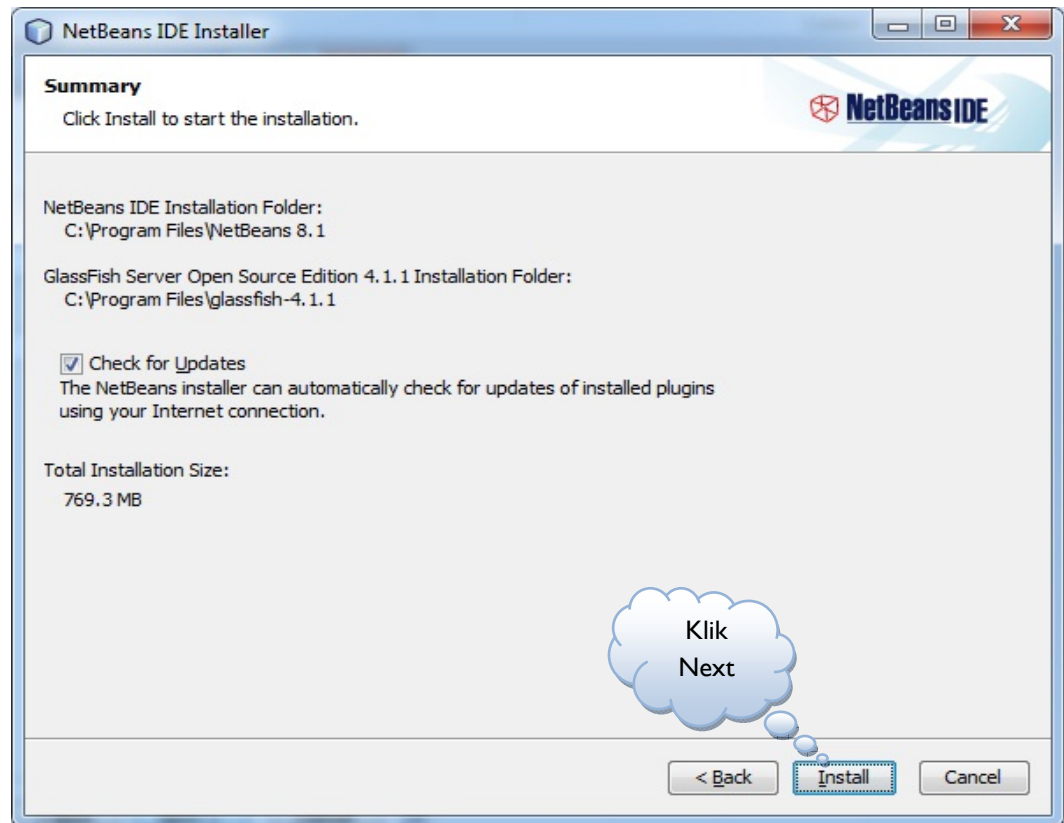
2. Maka akan muncul tampilan License Agreement berikut ini



Gambar 7. License Agreement

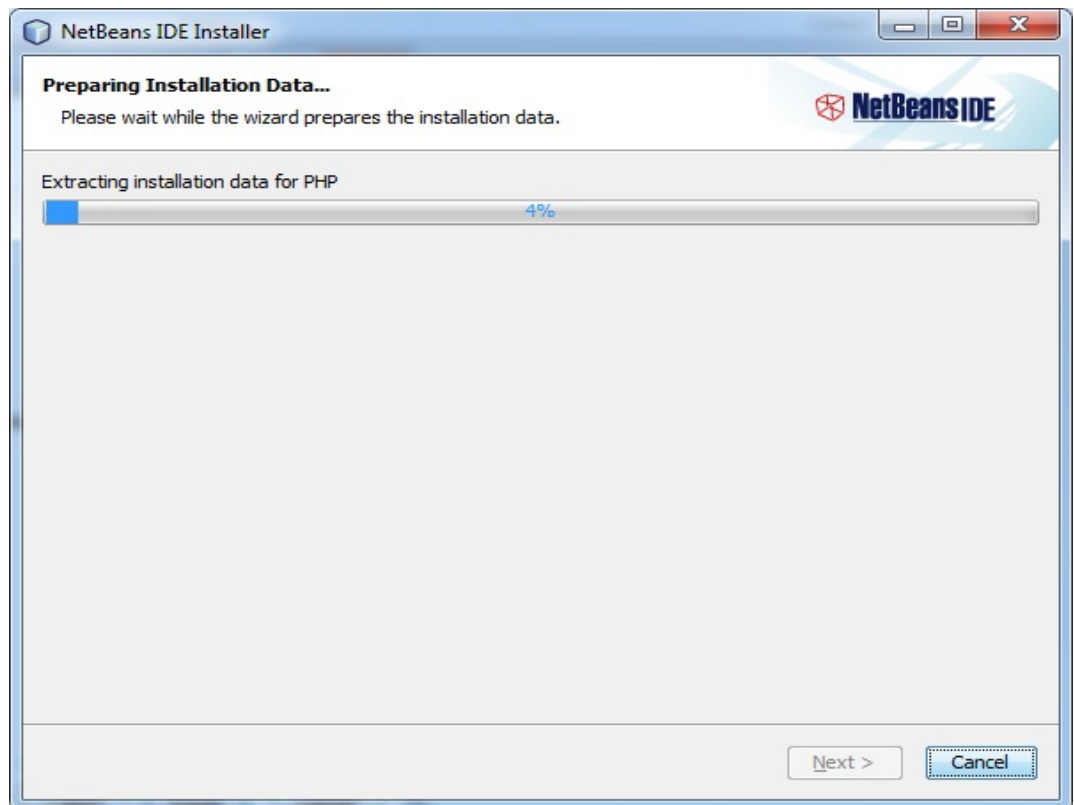


Gambar 8. Instalasi Folder

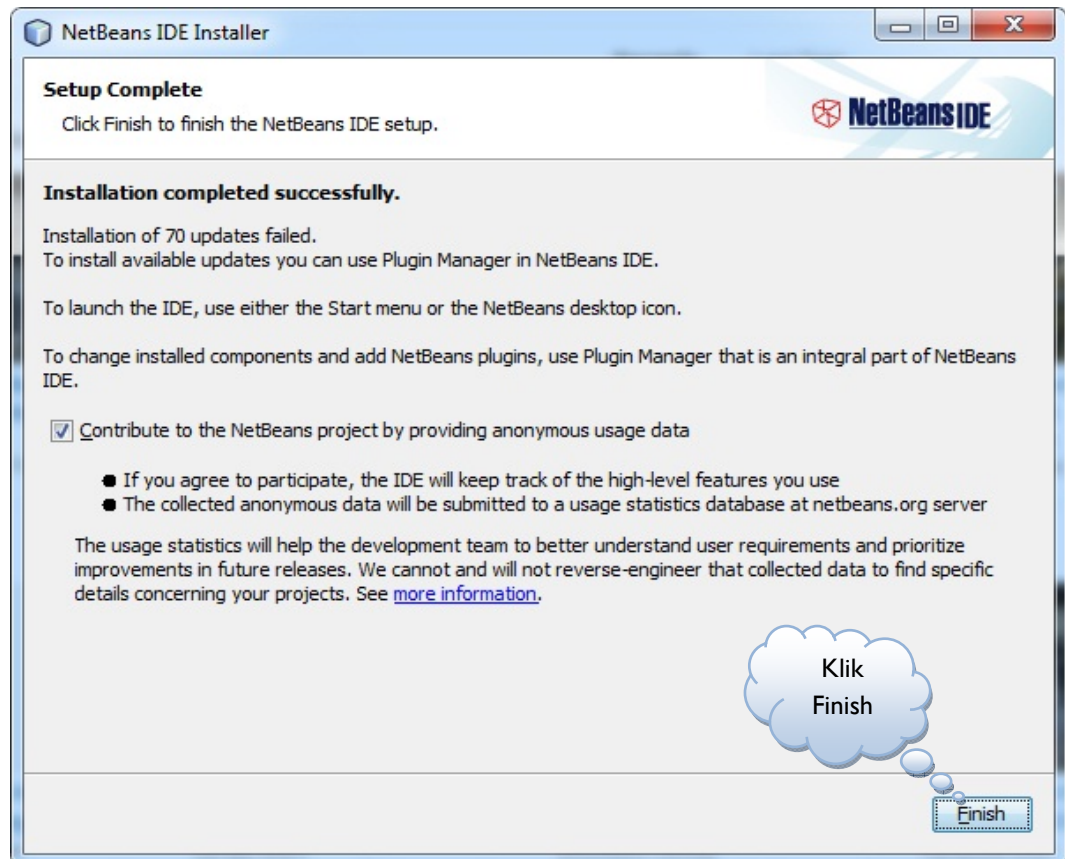


Gambar 9. Start the Installation

3. Proses instalasi berjalan seperti gambar dibawah ini.



Gambar 10. Proses Instalasi

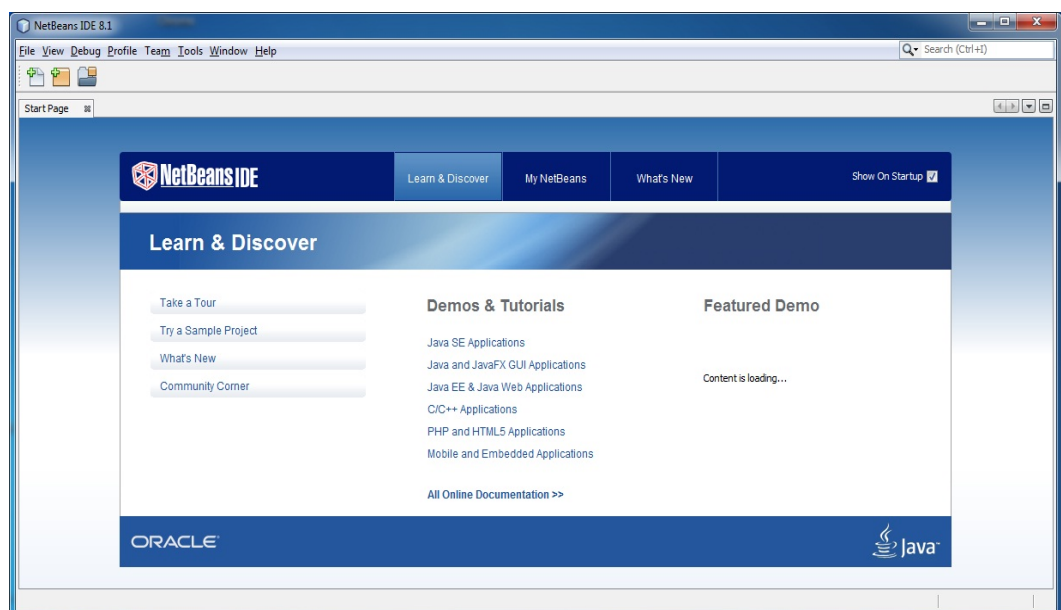


Gambar 10. Proses Instalasi Selesai

d. Membuat Project Baru

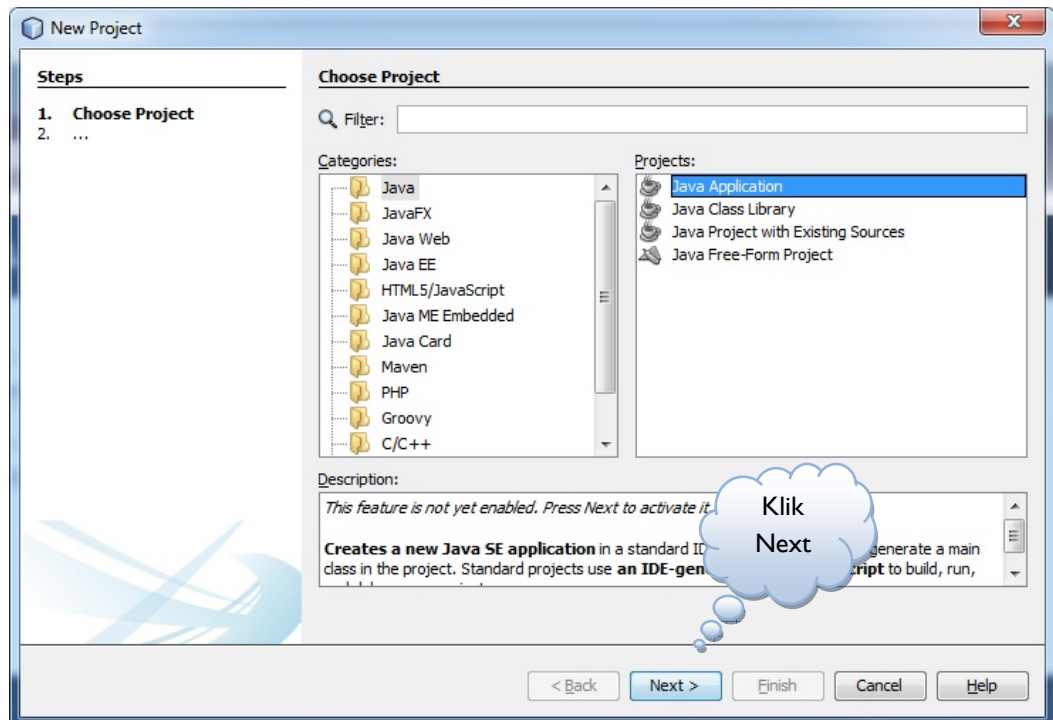
Langkah-langkah untuk membuat Project Baru :

1. Dari Start Program pilih Netbeans IDE 8.1



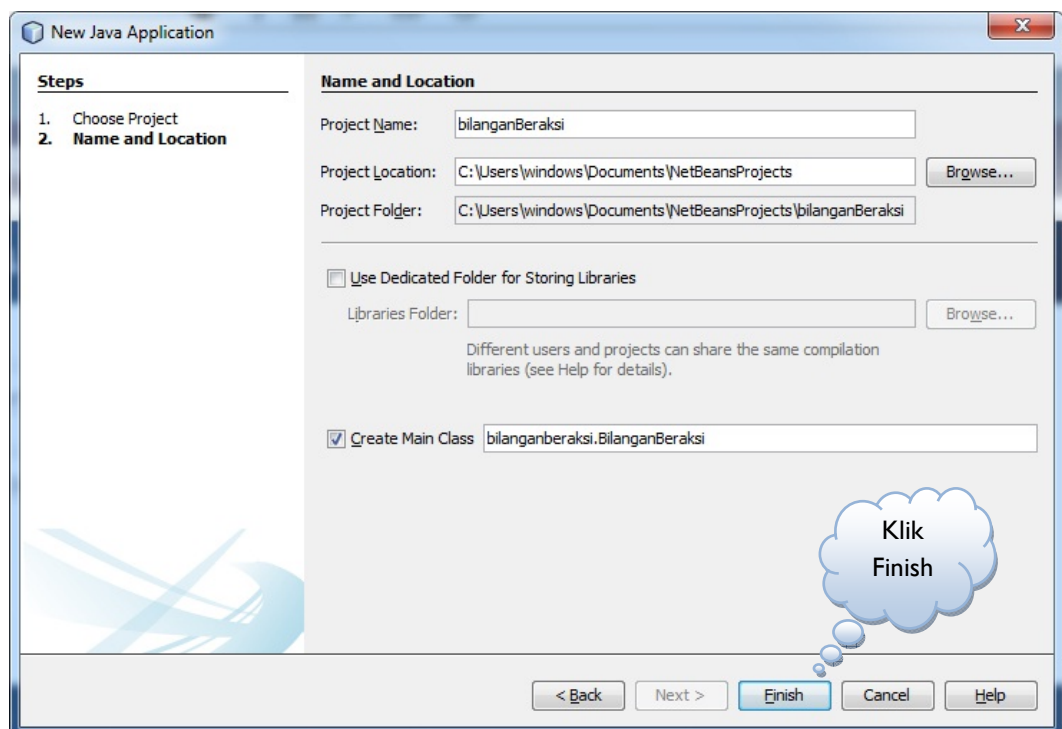
Gambar 11. Tampilan Awal NetBeans IDE

2. Pilih File → New Project → Java → Java Application



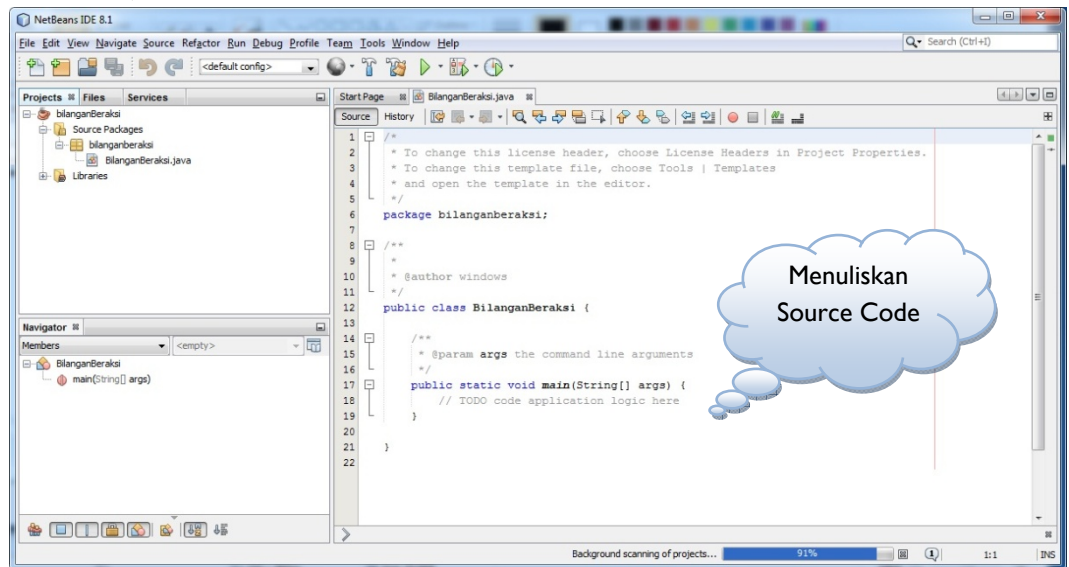
Gambar 12. New Project

3. Maka akan muncul Form dibawah ini untuk penamaan project dan lokasi penyimpanannya



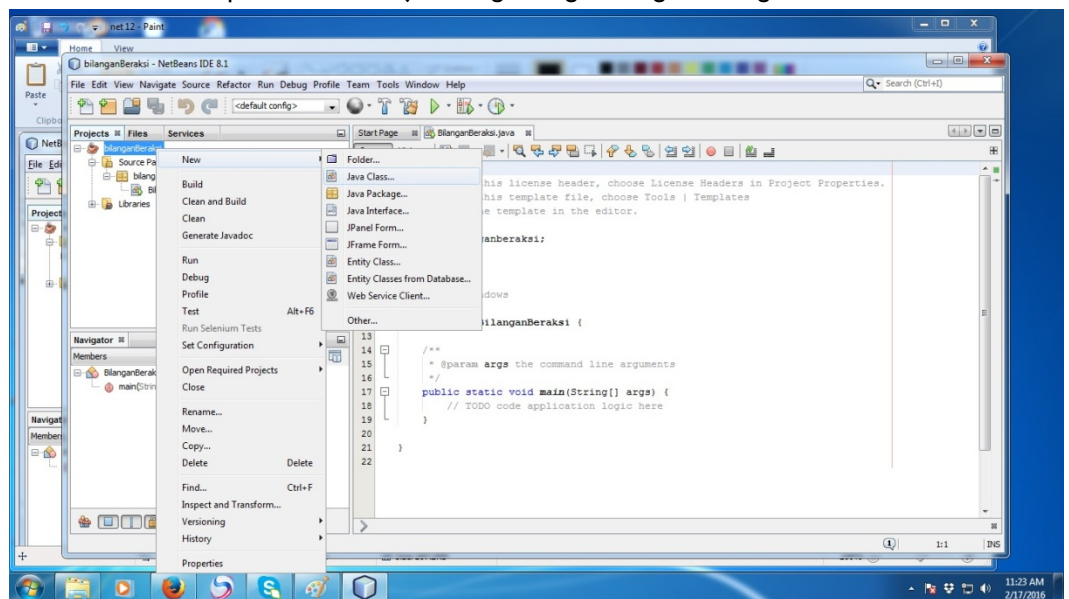
Gambar 13. Name and Location Project

4. Lembar kerja NetBeans untuk menuliskan Source Code

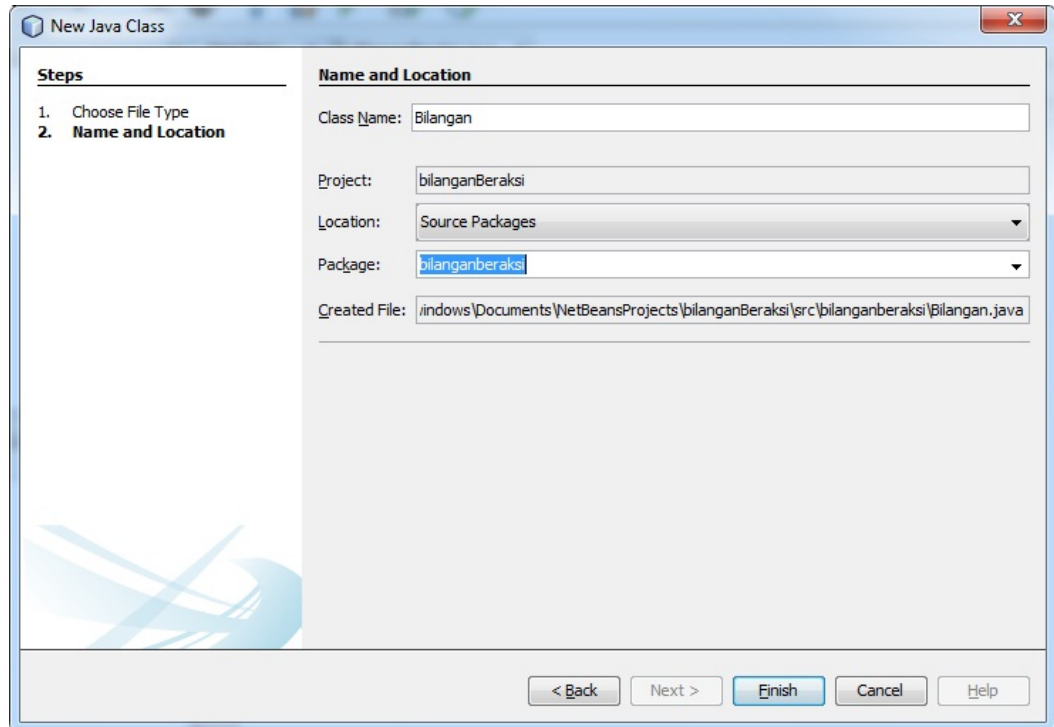


Gambar 14. Lembar kerja NetBeans

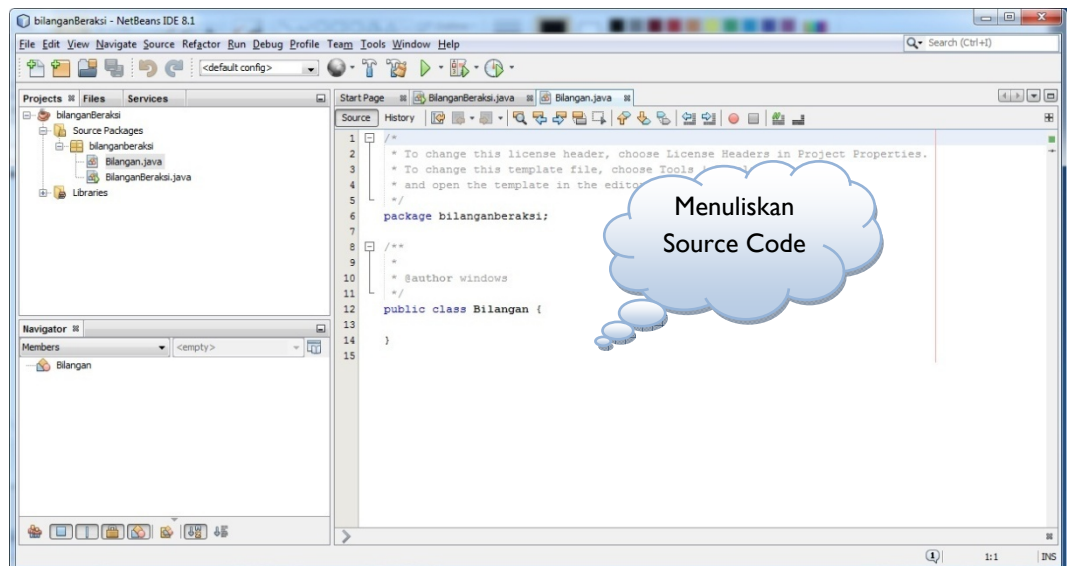
5. Menambahkan kelas pada sebuah Project dengan langkah-langkah sebagai berikut.



Gambar 15. Membuat Class



Gambar 16. Class Name



Gambar 16. Lembar Kerja Class

IV. GUIDED I

Membandingkan Program Terstruktur dan Program Berorientasi Objek

a. Pemrograman Terstruktur (Bahasa C++)

```
#include <stdio.h>
#include <stdlib.h>
#include <iostream.h>
#include <conio.h>

void helloWorld(int jmlh_looping)//Contoh Prosedur
{
    for(int i=0;i<jmlh_looping;i++)
    {
        cout<<i+1<<". Hello World..."<<endl;
    }
}

int tambah(int bil1,int bil2)//Contoh Fungsi
{
    return(bil1+bil2);
}

int main()
{

    int bil1=5,bil2=2;
    int jmlh_looping=5;
    cout<<endl;
    cout<<"-Display Hello World-"<<endl;
    helloWorld(jmlh_looping);//Pemanggilan Prosedur

    cout<<endl<<"-Pemanggilan Fungsi Tambah-"<<endl;
    //Pemanggilan Fungsi
    cout<<"Hasil Operasi Tambah : "<<tambah(bil1, bil2);
    cout<<endl;
    getch();
    return 0;
}
```

b. Pemrograman Berorientasi Objek (Bahasa C++)

```
#include <iostream>
#include <conio.h>

class DemoObjectOriented
{
public:
    void helloWorld(int jmlh_looping);
    int tambah(int bil1, int bil2);
};

void DemoObjectOriented::helloWorld(int jmlh_looping)
{
    for(int i=0;i<jmlh_looping;i++)
    {
        cout<<i+1<<". Hello World...."<<endl;
    }
}

int DemoObjectOriented::tambah(int bil1, int bil2)
{
    return(bil1+bil2);
}

int main()
{
    int bil1=5, bil2=2;
    int jmlh_looping=5;
    DemoObjectOriented Demo; //Pembentukan Objek
    cout<<endl;
    cout<<"-Display Hello World-"<<endl;
    Demo.helloWorld(jmlh_looping); //Pemanggilan Prosedur
    cout<<endl<<"-Pemanggilan Fungsi Tambah-"<<endl;
    //Pemanggilan Fungsi
    cout<<"Hasil Operasi Tambah : "<<Demo.tambah(bil1,bil2);
    cout<<endl;
    getch();
    return 0;
}
```

c. Pemrograman Berorientasi Objek (Bahasa Java)

1. Buat Project dengan nama **DemoApp**
2. Buat Class dalam Project tersebut dengan nama **DemoObjectOriented**
3. Dalam file class **DemoObjectOriented.java** tuliskan code berikut ini

```
import java.io.*;
class DemoObjectOriented
{
    public void helloWorld(int jmlh_looping)
    {
        System.out.print ("\n-Display Hello World-\n");
        for(int i=0;i< jmlh_looping;i++)
        {
            System.out.print (+(i+1)+". Hello
            World...\n");
        }
    }
    public int tambah(int bill1, int bil2)
    {
        return(bill1+bil2);
    }
}
```

4. Dalam file **class Main.java** tuliskan code berikut ini

```
public static void main(String args[])
{
    int jumlah_looping=5;
    int bill1=5;
    int bil2=2;
    DemoObjectOriented Demo = new DemoObjectOriented ();
    Demo.helloWorld(jumlah_looping);
    System.out.print("\n-Pemanggilan Fungsi Tambah-\n");
    System.out.print("        Hasil        Operasi        Tambah        :
    "+Demo.tambah(bill1,bil2)+"\n");
}
```

V. GUIDED 2

1. Membuat kelas untuk menampilkan kalimat "Selamat Datang di Praktikum PBO di dalam C++ dan Java

```
#include <iostream>
#include <conio.h>

class Welcome
{
    public:
    void display()
    {
        cout<<" Selamat Datang di Praktikum PBO "<<endl;
    }
};

void main()
{
    Welcome W;
    W.display();
    getch();
}
```

Bentuk 1

```
#include <iostream>
#include <conio.h>

class Welcome
{
    public:
    void display();
};

void Welcome::display()
{
    cout<<" Selamat Datang di Praktikum PBO "<<endl;
}

void main()
{
    Welcome W;
    W.display();
    getch();
}
```

Bentuk 2

Amati perbedaan kedua program di atas !! Bahas dalam Laporan !!

Welcome.java

```
public class Welcome
{
    public void display()
    {
        System.out.println("Selamat Datang di Praktikum PBO");
    }
}
```

Main.java

```
public static void main(String []args)
{
    Welcome W=new Welcome ();
    W.display();
}
```

2. Membuat Fungsi Input dan Output dalam C++

```
class Mahasiswa
{
    private:
        char nama[20];
        char nim[20];

    public:
        void inputData();
        void display();
};

void Mahasiswa::inputData()
{
    cout<<"\n-Input Data Mahasiswa-<<endl;
    cout<<"Masukkan Nama Mahasiswa : ";
    cin>>nama;
    cout<<"Masukkan Nomor Induk Mahasiswa : ";
    cin>>nim;
}

void Mahasiswa::display()
{
    cout<<"\n-Display Data Mahasiswa-<<endl;
    cout<<"Nama : "<<nama<<endl;
    cout<<"NIM : "<<nim<<endl;
};

void main()
{
    Mahasiswa mhs;
    mhs.inputData();
    mhs.display();
    getch();
}
```

Apakah program ini dapat dijalankan? Jika tidak, temukan alasannya dan Bahas dalam laporan !!

3. Membuat Fungsi Input dan Output beserta Looping (perulangan) dalam Java

Mahasiswa. java

```
import java.io.*; //Pemanggilan Library dalam Java
class Mahasiswa
{
    private String Nama[]= new String[3]; //Deklarasi array
                                           // dalam Java
    private String NIM[]= new String[3];
    private int i;
    public void inputData()
    {
        BufferedReader b;
        b=new BufferedReader(new InputStreamReader
        (System.in));
        try //Penjelasan Exception lebih lanjut di berikutnya
        {
            System.out.println("-Input Data Mahasiswa-");
            for (i=0;i<3;i++) // Perulangan (looping)
            {
                System.out.print ("Masukkan Nama : ");
                Nama[i] =b.readLine (); //Pembacaan
                inputan melaluumkeyboard (seperti "cin" dalam C++)
                System.out.print ("Masukkan NIM : ");
                NIM[i] =b.readLine ();
            }
        }
        catch (Exception E){}
    }
    public void display()
    {
        System.out.println("");
        System.out.println("-Display Data Mahasiswa-");
        for (i=0;i<3;i++)
        {
            System.out.println("Nama : "+Nama[i]);
            System.out.println("NIM : "+NIM[i]);
            System.out.println("");
        }
    }
}
```

Main.java

```
public static void main(String args[])
{
    Mahasiswa M=new Mahasiswa(); //Pembentukan Objek
    M.inputData(); // Pemanggilan method
    M.display ();
}
```